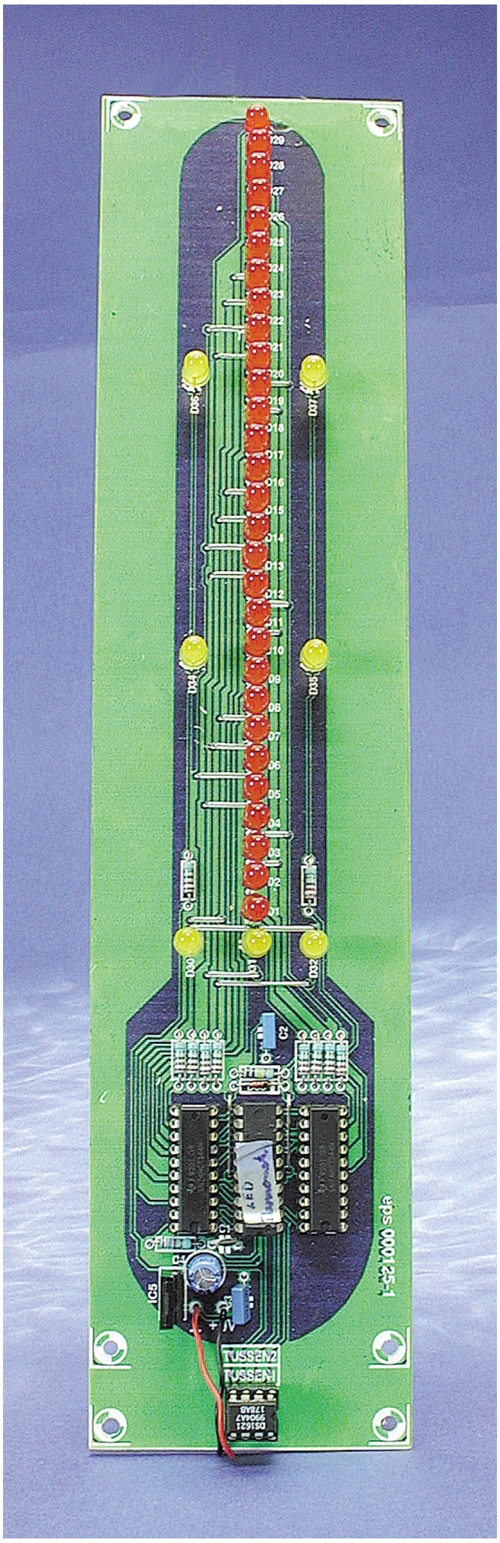
LED Thermometer

Dallas sensor IC and COP8 microcontroller



By F. Wohlrabe

The project presented here provides a novel form of temperature display. Its appearance makes it an ideal add-on to the ‘Rhine Tower Clock’ and will certainly attract attention.

At the heart of the circuit is a National Semiconductor COP8782 microcontroller, as has already been used in *Elektor Electronics* projects like the Running Text Display (PC Topics Supplement, February 2000). The microcontroller reads temperature data from a sensor IC made by Dallas Semiconductor, which has also been seen before in *Elektor Electronics* projects (‘Temperature Measurements with the DS1621’ and ‘DS1621 Datasheet’, both in the March 2000 issue). A DS1621 programmer was published in the February 2001 issue.

The microcontroller also has the job of displaying the measured value on 32 multiplexed LEDs.

TEMPERATURE SENSOR

Since the principles of operation of the DS1621 have already been described in Elektor Electronics, we can concentrate here on the important features. The sensor incorporates a complete temperature measurement system, including a thermostat function. All data are communicated over an I^{2}C bus. The sensor operates from –55 °C to +125 °C, with a resolution of 0.5 °C. In this project, the display is limited to the range –29.5 to +59.5 °C. The thermostat function of the DS1621 (output T_{out}) is, as can be seen in the circuit diagram in **Figure 1**, not used in this application.

Technical Data

Microprocessor controlled	
Temperature range	–29.5 to +59.5 °C
Resolution	0.5 °C
Error over temperature range	±0.5 °C
Supply voltage	9 V
Supply current	approx. 100 mA

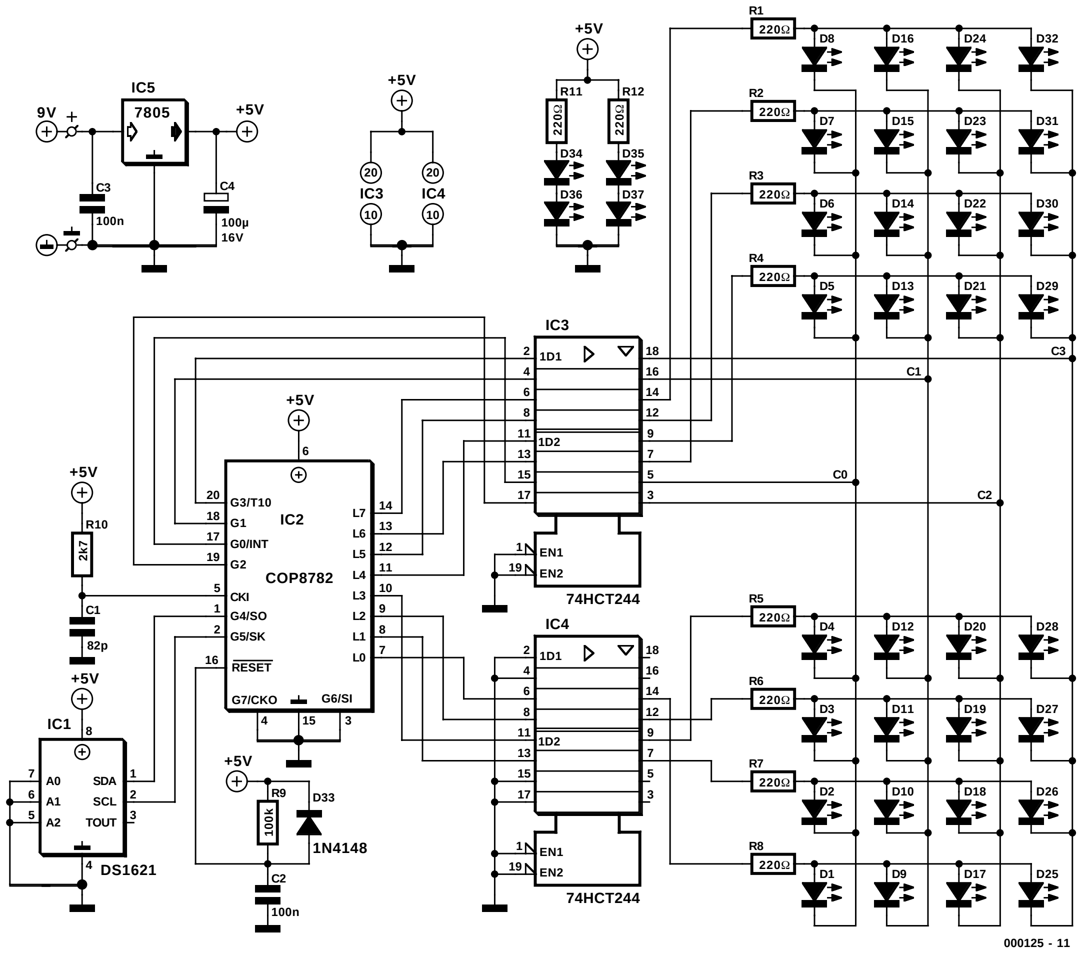


Figure 1. Circuit diagram of the LED thermometer: 32 LEDs driven from 12 I/O ports of a COP microcontroller.

In theory up to eight sensors can be connected to the same bus, each device being assigned a different address using the address inputs A0, A1 and A2. For this LED thermometer we do not need this feature, so all the address inputs are tied to ground and the sensor is thereby assigned address zero.

The temperature is read out in a 9-bit format, arranged as follows:

Temp.	Output data	
in °C	binary	hex
+125	01111101 00000000	7B00
+25	00011001 00000000	1900
+0,5	00000000 10000000	0080
0	00000000 00000000	0000
- 0,5	11111111 10000000	FF80
- 25	11100111 00000000	E700
- 55	11001001 00000000	C900

Byte 1 contains the integer part of the temperature, with a negative value indicated by the MSB being set. The MSB of byte 2 being set indicates an extra +0.5 °C. The remaining bits of byte 2 are always zero.

According to the data sheet, each measurement takes at most 1 s. In order to read the temperature value out of the sensor, the microcontroller issues the following commands:

Microcontroller commands

Once only, to configure sensor:

1001 0000	4 bit control code plus device address, write mode
1010 1100	write to configuration register
0000 0010	continuous measurements

Before each measurement:

1001 0000	4 bit control code plus device address, write mode
1110 1110	‘start measurement’ command

After a 1 s delay:

1001 0000	4 bit control code plus device address, write mode
1010 1010	‘read temperature’ command
1001 0001	4 bit control code plus device address, read mode

Read in data as shown in data sheet timing diagram

The typical error in the reading from the DS1621 is shown in **Figure 2**. In the range of interest to us it is at most 0.5 °C.

SENSOR AND COP

The COP8782 microcontroller from National Semiconductor has many features that make it particularly well suited to this project:

- 4096 x 8 (OTP) EPROM
- 128 bytes RAM
- 1 µs cycle time at 10 MHz
- 16 bit timer with automatic reload, external event counter and capture functions
- 16 I/O pins, 14 of which can be individually programmed as inputs or outputs
- Selectable tri-state, push-pull or pull-up I/O pin configuration
- Microwire interface
- Timer, software, and external interrupt sources, with programmable polarity

The COP8AC7 is a broadly pin- and functionally-compatible successor device to the COP8782. The starter kit unfortunately does not offer real-time emulation, but it does allow programming of OTP devices and offers a comprehensive introduction to this economical and technically interesting family of microcontrollers.

For more demanding projects, where the real-time performance of the microcontroller must be verified, the purchase of an emulator is essential or else the project will degenerate into continuous experimentation.

DRIVER AND DISPLAY

In order to minimise the number of connections for a large number of display elements, the 32 LEDs are driven in a multiplexed fashion. Four LEDs (D34-D37) are permanently lit (reminiscent of the Rhine Tower Clock) and indicate the tens of degrees, making the display easier to read. Since pins G6 and G7 can only be used as inputs, the other twelve ports are all used as outputs. With the use of clever software we can drive 32 LEDs through these 12 ports. The LED matrix is divided into four groups (A-D), each consisting of eight LEDs. These groups are driven from port pins G0-G3 in sequence, so that at most eight LEDs are ever lit simultaneously. The LEDs are turned on by setting port pins L0-L7 high. With a low level on one of G0-G3 the appropriate group of LEDs is enabled. Although only at most eight LEDs are on at any one time, the multiplexing happens so fast that the visual appearance is that up to 32 LEDs are lit.

The overall temperature range from –29.5 °C to +59.5 °C is divided into three ranges (-29 to 1 °C, 0 to+29 °C and 30 to 59 °C), as indicated by the three range LEDs (D30, D31 and D32). The odd 0.5 °C is indicated by the range LED flashing at about 1 Hz. So, for example, at 20.5 °C the middle range LED D31 flashes, indicating a temperature between 0 °C and 29 °C.

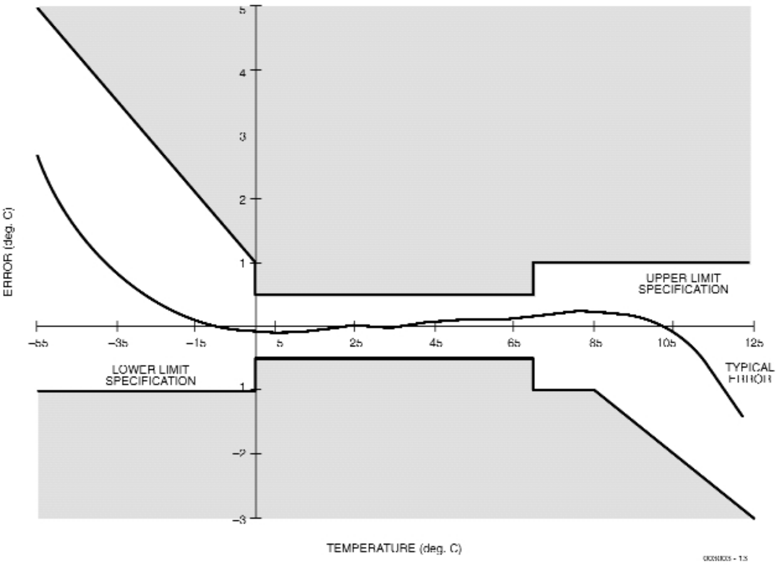
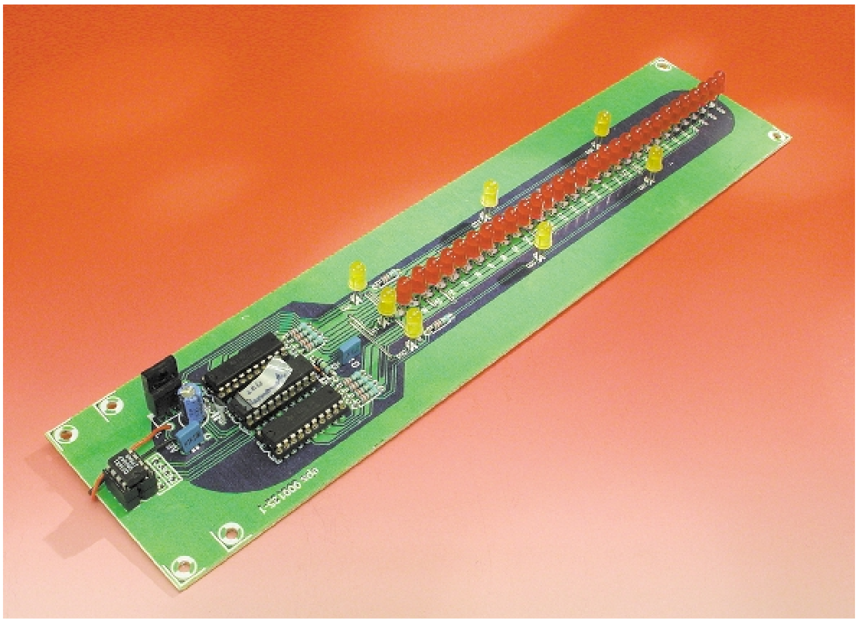


Figure 2. Typical error characteristic of the DS1621.

CONSTRUCTION

Since the outputs of the microcontroller cannot provide enough current to drive the high efficiency LEDs directly, the twelve drive signals are buffered using a 74HCT244 driver. Because of the high multiplex ratio and the drive impedance of the ports, current-limiting resistors of only 220 Ω can be used. Maximum brightness (although with a current consumption of 180 mA!) can be obtained by dispensing with current-limiting resistors altogether, substituting wire links. An ordinary (unstabilised) mains adaptor, with an output voltage of at least 9 V, can be used as a power supply.

Assembling the circuit board (**Figure 3**) should present no difficulties. The sensor device can be separated from the rest of the circuit, allowing remote mounting if for example it is desired to measure the temperature outside. The I²C bus and power supply can be connected using a cable at most 3 m long.



Software

Circuit board layout, source and object code can be downloaded from the Free Downloads page of the Elektor Electronics website. Source and object code are also available from Readers' Services on disk number 000125-11, and a ready-programmed microcontroller (000125-41) is also available.

For a tidy appearance a template should be used when bending the resistors and wire links for fitting to the circuit board. When soldering, use a spacer to ensure that the LEDs are accurately lined up. Check the polarity of the LEDs: an incorrectly fitted LED may be hard to track down.

(000125-1)

COMPONENTS LIST

Resistors:

R1-R8,R11,R12 = 220Ω
R9 = 100kΩ
R10 = 2kΩ7

Capacitors:

C1 = 82pF
C2,C3 = 100nF
C4 = 100µF 16V radial

Semiconductors:

D1-D29 = LED, high efficiency, red
D30,D31,D32 = LED, high efficiency, yellow
D33 = 1N4148
IC1 = DS1621 (Dallas Semiconductor) (Farnell # 760-704)
IC2 = COP8782-CN, programmed, order code **000125-41**
IC3,IC4 = 74HCT244
IC5 = 7805

Miscellaneous:

PC1,PC2 = solder pins
PCB, order code **000125-1**
Disk, COP8 source code file, order code **000125-11**

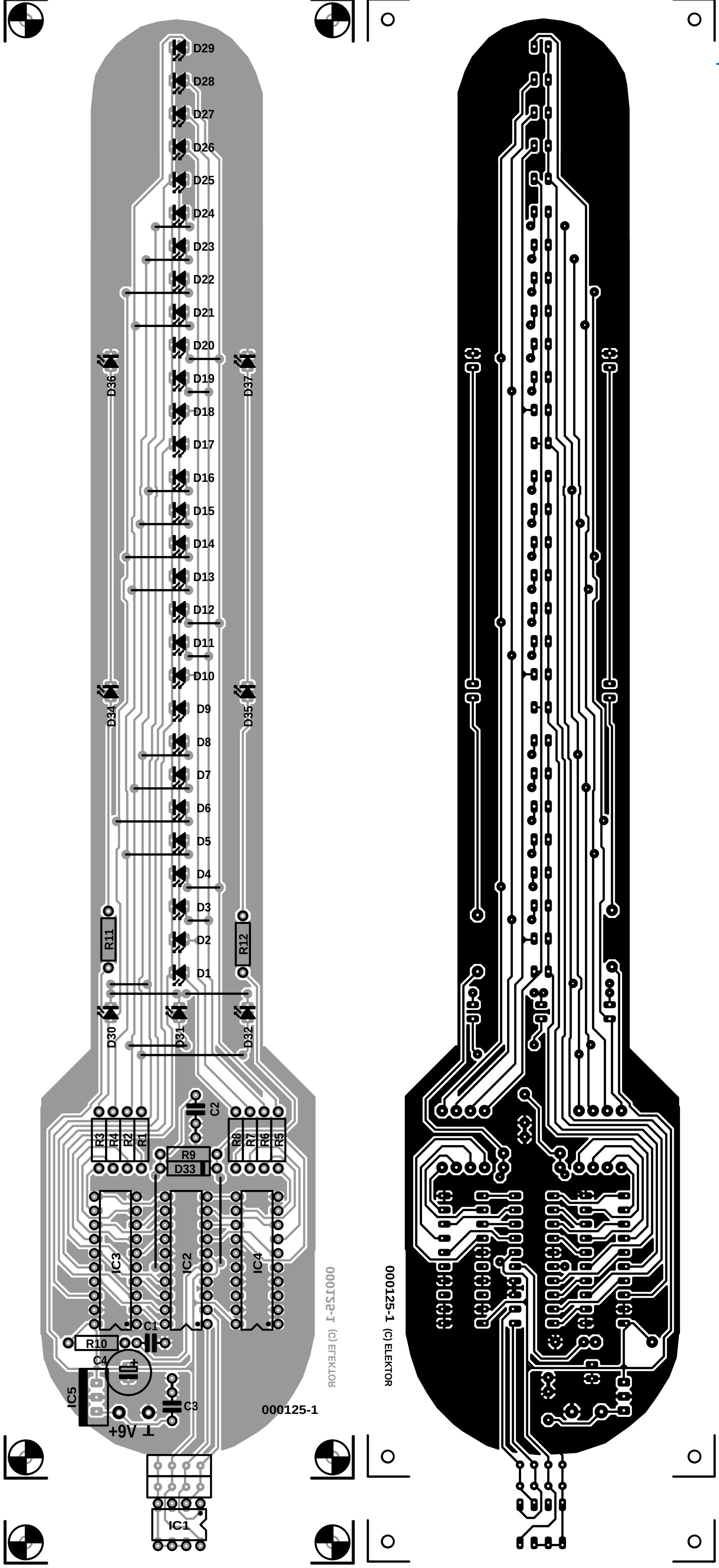


Figure 3. Circuit board, shaped similarly to that for the Rhine Tower Clock.